

## ЛАБОРАТОРНАЯ РАБОТА №6

### ДИНАМИЧЕСКИЕ МАССИВЫ. ЦИКЛИЧЕСКИЕ ДЕЙСТВИЯ С ОБЪЕКТАМИ

Если заранее неизвестно, сколько элементов будет в массиве, можно использовать динамический массив. При объявлении такого массива указывается только тип данных, для которых он создается, а индексы не указываются:

**Dim A() as Long** *'объявлен динамический массив целых чисел*

Перед использованием динамического массива его размерность должна быть определена посредством инструкции ReDim.

При использовании инструкции ReDim создается массив указанного размера, при этом имевшиеся ранее в элементах значения не сохраняются:

**ReDim A(5)** *'Определение размера массива*  
**For N=0 To 5** *'Начало цикла заполнения массива*  
**M1(N)=2\*N+1** *'Очередной элемент*  
**Next N** *'Конец цикла*  
**'ReDim M1(10)** *'Массив расширяется, данные теряются*

Для того, чтобы имеющиеся в массиве данные при изменении его размера не терялись, следует добавить слово Preserve:

**ReDim Preserve A(10)** *'Массив изменяется  
'с сохранением имеющихся в нем данных.*

Если требуется выполнить группу действий для каждого объекта из коллекции (коллекция – это набор однотипных объектов: ячеек в диапазоне, листов в книге и т.п.), существует специальный оператор:

**For Each** *объект* **In** *коллекция*

*Операторы*  
**Next** объект

При этом тип объекта, имя которого указывается после слова **Each**, должен соответствовать типу объектов в коллекции. То есть *коллекция* в данном случае – это набор объектов соответствующего типа. При назначении типа данных для *объекта* это необходимо учитывать. Можно также для *объекта* указать тип данных **Object** (объект любого типа).

Пример

Требуется заполнить диапазон ячеек A1:C9 числами от 1 до 9.

**Dim cl As Range, I As Byte**

**K=1**

**For Each cl In Range("A1:C9")**

**cl.value=K**

**K=K+1**

**Next cl**

Обратите внимание, что в этом примере переменная *cl* объявлена как имеющая тип **Range** (диапазон), а коллекция после слова **In** – это тоже диапазон ячеек. Таким образом, в заголовке цикла организуется перебор всех ячеек в диапазоне (одиночная ячейка тоже является диапазоном). При этом можно было для переменной *cl* указать тип данных **Object**.

Если заранее неизвестно, в каком диапазоне на листе введены данные, можно использовать объект **ActiveSheet.UsedRange**. Он представляет из себя диапазон, начиная от первой до последней непустой ячейки на листе. Для того, чтобы узнать номер первой заполненной строки следует вызвать свойство **.Rows(1).Row** для этого объекта. Чтобы определить номер первого заполненного столбца следует вызвать свойство **.Columns(1).Column**.

Задача 3.1. На листе, начиная с любой ячейки, в строку введены произвольные целые числа (как положительные, так и отрицательные). Составить программу для вывода в порядке возрастания всех положительных чисел в одной строке, а

отрицательных в другой (ноль считать положительным). Вывести также исходный массив в порядке возрастания значений.

Решение

Сначала определим и запомним в соответствующих переменных номер первой заполненной строки и номер первого заполненного столбца, используя диапазон `ActiveSheet.UsedRange`. Затем из этого диапазона прочитаем в первый массив все имеющиеся в строке данные. После этого отсортируем этот массив.

Затем “пройдем” по этому массиву от первого до последнего элемента и, если встретим отрицательный элемент, запишем его во второй массив, в противном случае элемент запишем в третий массив. Учитывая, что исходный массив был отсортирован по возрастанию, получим массивы отрицательных и положительных чисел тоже отсортированными. Внутри этого “прохода” можно также вывести первый массив в отсортированном виде ниже исходного.

Наконец, выведем второй и третий массивы. Учитывая то, что в исходном массиве могло и не оказаться положительных или отрицательных чисел, прежде, чем выводить элементы массива, проверим номер последнего индекса соответствующего массива и, если он нулевой, выведем сообщение об отсутствии соответствующих чисел.

Пример программы с необходимыми комментариями приведен ниже.

**Public Sub Primer3\_1()**

**Dim UR As Range, CL As Range** *'диапазон ячеек с данными и ячейка в нем*

**Dim R1 As Byte, C1 As Byte, R As Byte**

**Dim M1() As Integer, M2() As Integer, M3() As Integer**

**Dim i As Byte, j As Byte, k As Byte** *'счетчики количеств элементов массивов*

**Dim N As Byte** *'Счетчик цикла и номер элемента массива*

**Dim Sort As Boolean** *'Признак перестановки при сортировке*

**Dim Buf As Integer** *'Буфер для обмена элементов массива данными*

**i = 0**

```

Set UR = ActiveSheet.UsedRange
R1 = UR.Rows(1).Row 'запоминаем номер первой строки с данными
C1 = UR.Columns(1).Column 'запоминаем номер первого столбца
'с данными
'Заполняем первый массив данными с листа
For Each CL In UR
    i = i + 1
    ReDim Preserve M1(1 To i)
    M1(i) = CL.Value
Next CL
'сортировка исходного массива
'сейчас переменная i хранит номер последнего элемента массива
Sort = True
'Начало сортировки
While Sort
    Sort = False
    For N = 1 To i - 1
        If M1(N) > M1(N + 1) Then 'Перестановка
            Buf = M1(N)
            M1(N) = M1(N + 1)
            M1(N + 1) = Buf
            Sort = True 'Признак произошедшей перестановки
        End If
    Next N
Wend
'элементы первого массива расположены по возрастанию
Cells(R1 + 1, C1) = "Результат сортировки" 'Вывод заголовка
'Формирование массивов положительных и отрицательных чисел
j = 0
k = 0
For N = 1 To i
    If M1(N) < 0 Then
        j = j + 1
        ReDim Preserve M2(1 To j)
        M2(j) = M1(N)
    Else

```

```

    k = k + 1
    ReDim Preserve M3(1 To k)
    M3(k) = M1(N)
End If
Cells(R1+2, C1+N - 1) = M1(N) 'Вывод элемента первого массива
Next N
'Вывод второго массива
If j > 0 Then
    Cells(R1 + 3, C1) = "Отрицательные числа" 'Вывод заголовка
    For N = 1 To j
        Cells(R1+4, C1+N-1)=M2(N) 'Вывод элемента второго массива
    Next N
Else
    Cells(R1 + 3, C1) = "Отрицательные числа не обнаружены"
End If
'Вывод третьего массива
If k > 0 Then
    Cells(R1 + 5, C1) = "Положительные числа" 'Вывод заголовка
    For N = 1 To k
        Cells(R1+6, C1+N-1)=M3(N) 'Вывод элемента второго массива
    Next N
Else
    Cells(R1 + 5, C1) = "Положительные числа не обнаружены"
End If
End Sub

```

В данной программе для работы с заполненным диапазоном был использован объект UR, что позволило несколько сократить размер записей для обращения к данным и свойствам этого объекта.

Для сортировки массива использовался так называемый метод “пузырька”. Суть метода в следующем.

Выполняем “проходы”, внутри каждого из которых сравниваем очередной элемент, начиная с первого, со следующим элементом. Если следующий элемент окажется меньше текущего, переставляем их местами. Для временного хранения одного из значений при перестановке использована переменная Buf. Добравшись до предпоследнего элемента массива и сравнив его с

последним, проверяем, состоялась ли за данный “проход” хотя бы одна перестановка. Для этого в программе используется переменная Sort логического типа. Перед началом очередного “прохода” ей присваивается значение False (Ложь), а после любой перестановки она принимает значение True (Истина). Таким образом, “проходы” будут выполняться до тех пор, пока переменная Sort имеет значение True (цикл While Sort ...Wend).

Результат работы программы показан на рисунке 3.1.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
3			5	-8	17	25	38	-14	18	-2	11	-10
4			Результат сортировки									
5			-14	-10	-8	-2	5	11	17	18	25	38
6			Отрицательные числа									
7			-14	-10	-8	-2						
8			Положительные числа									
9			5	11	17	18	25	38				
10												

Рисунок 3.1

### ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задача 3.1. На листе, начиная с любой ячейки, в строку введены произвольные целые числа (как положительные, так и отрицательные). Составить программу для вывода в указанном порядке всех чисел в соответствии условием 1 в одной строке, а в другой - всех чисел в соответствии условием 2. Вывести также исходный массив в порядке возрастания значений и в порядке убывания значений.

Вариант	Условие 1	Условие 2
1	четные положительные	нечетные
2	делящиеся на 3 положительные	четные
3	нечетные	положительные
4	нечетные отрицательные	четные положительные
5	четные отрицательные	нечетные положительные

6	делящиеся на 3 отрицательные	четные положительные
7	неделящиеся на 3	нечетные отрицательные
8	делящиеся на 5 отрицательные	нечетные
9	делящиеся на 3 четные	делящиеся на 3 нечетные
10	делящиеся на 3 положительные	делящиеся на 3 отрицательные